BiMoGen: Bidirectional Motion-Text Generation via Unified Masked Discrete Diffusion

Wanjiang Weng[1,2,*] **Yongliang Wu**[1,2,*] **Xiaofeng Tan**[1,2]
Xingyu Zhu[3] **Wenbo Zhu**[4] **Hongsong Wang**[1,2,†]
[1]Department of Computer Science and Engineering, Southeast University, Nanjing, China
[2]Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, Nanjing, China
[3]National University of Singapore
[4]HeyGen

Abstract

Text-to-motion generation and motion-to-text captioning are two fundamental tasks in human motion modeling, both grounded in the same underlying motion-text correspondence. Existing unified approaches mostly rely on autoregressive modeling, which imposes a fixed generation order and is therefore poorly suited to the bidirectional dependencies between language and motion, allowing early prediction errors to persist as fixed context and degrade both temporal coherence and cross-modal consistency. Masked discrete diffusion, which models sequences through iterative bidirectional prediction, allows prediction from bidirectional context without a fixed generation order. We therefore propose **BiMoGen** (**Bi**directional **Mo**tion-text **Gen**eration), a unified masked discrete diffusion framework for bidirectional motion-text modeling. To stabilize training, we design Decoupled Uni- and Cross-Modal Training, in which masked pretraining first establishes cross-modal correspondence on paired motion-text sequences, after which supervised fine-tuning specializes the model for bidirectional generation. Masked diffusion nonetheless faces a train-test mismatch, as the model is trained on clean ground-truth context yet encounters self-generated and potentially erroneous context at inference, with errors committed under heavily masked states propagating through subsequent steps. We further introduce Generation-Aware Self-Correction that exposes the model to its own predictions during training and uses correction passes at early sampling steps for further refinement of previously unmasked tokens. Extensive experiments on HumanML3D and KIT-ML demonstrate competitive performance on both tasks, validating the effectiveness of the proposed two-stage training and self-correction designs. The project page is available at `https://wengwanjiang.github.io/BiMoGen-Page`.

1 Introduction

Text-to-motion generation (T2M) and motion-to-text captioning (M2T) are two closely related tasks that both rely on understanding the correspondence between human motion and natural language [57, 51, 27, 42, 44]. Recent work has therefore pursued unified frameworks that address both tasks within a single model, so that both conditional directions can use the same backbone and motion-text token space. Many existing approaches build on autoregressive modeling [21, 75, 53, 74]. Token-based approaches discretize motion, concatenate motion and text tokens, and train a Transformer through next-token prediction [21, 53, 29]. Yet the underlying causal backbone imposes a fixed left-to-right generation order and provides no mechanism to revise early prediction errors. Once a token is

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

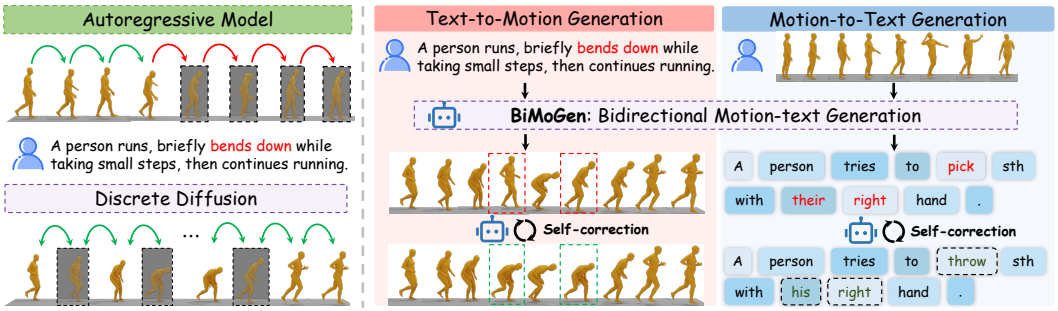

Figure 1: Overview of the BiMoGen framework. BiMoGen unifies text-to-motion generation and motion-to-text captioning within a single masked discrete diffusion model, and employs a self-correction mechanism that revises unreliable predictions at early sampling steps.

incorrectly produced, it remains in the prefix conditioning all subsequent steps, and the resulting errors propagate through the sequence and ultimately degrade temporal coherence and motion-text consistency.

Masked discrete diffusion offers a more principled alternative [60, 31, 2, 35, 4, 40, 1]. By iteratively predicting masked tokens from bidirectional context, it naturally captures the bidirectional dependencies between text and motion without imposing a fixed left-to-right generation order. However, applying masked diffusion to unified motion-text modeling raises two fundamental challenges. First, training a single model for both T2M and M2T entangles cross-modal correspondence learning with conditional generation, forcing the model to produce target tokens before reliable motion-text alignment has emerged and destabilizing optimization. Second, the model is trained on corrupted ground-truth context yet must condition on its own earlier predictions at inference, and this train-test misalignment becomes most severe under the high mask ratios encountered in early sampling steps, where committed errors persist and cascade into later predictions.

To address these challenges, we propose **BiMoGen** (**Bi**directional **Mo**tion-text **Gen**eration), a unified masked discrete diffusion framework for bidirectional motion-text modeling. As shown in Figure 1, BiMoGen handles both T2M and M2T within a single model through masked token prediction, naturally accommodating the bidirectional dependencies between the two modalities. Recognizing that cross-modal alignment and conditional generation impose fundamentally different learning objectives, we introduce a two-stage strategy named Decoupled Uni- and Cross-Modal Training. The first stage performs masked pretraining on paired motion-text sequences to establish robust cross-modal correspondence, while the second stage applies supervised fine-tuning that specializes the model for bidirectional generation. To further close the gap between training and inference, we propose Generation-Aware Self-Correction that explicitly aligns the two regimes. During training, the model learns to recover ground-truth tokens from its own predicted context, while at inference correction passes at early sampling steps provide further refinement of previously unmasked tokens. Extensive experiments on HumanML3D [13] and KIT-ML [38] demonstrate that BiMoGen achieves competitive performance on both T2M generation and M2T captioning.

Our contributions are summarized as follows:

- We propose BiMoGen, a unified masked discrete diffusion framework that formulates T2M generation and M2T captioning under a shared bidirectional architecture.
- We introduce Generation-Aware Self-Correction, applied during training to mitigate the train-test misalignment inherent in masked diffusion and during sampling for further refinement.
- We design Decoupled Uni- and Cross-Modal Training, which comprises masked pretraining and supervised fine-tuning stages to disentangle cross-modal alignment from conditional generation.

2 Related Works

Text-to-Motion Generation. Generating 3D human motion from natural language has emerged as a fundamental task with applications in animation, virtual humans, and embodied agents [58, 71, 57, 56, 23, 24, 45, 7, 10]. Continuous-diffusion approaches model motion either in raw coordinate

space [33, 34, 39, 55, 79, 77, 78], as in MDM [46], or in a learned latent space for efficiency, as in MLD [6]. A parallel line of work discretizes motion into tokens via VQ-VAE [47] and models them with sequence models [37, 22, 41]. T2M-GPT [65] adopts a GPT-style autoregressive decoder, whereas MoMask [15] introduces residual quantization with masked generative modeling. BAMM [37] combines autoregressive generation with bidirectional refinement, while BiPO [18] uses partial occlusion and selective token regeneration. Their bidirectional prediction is trained on ground-truth context, whereas GASC explicitly trains recovery from self-generated context. Despite the diversity of generative paradigms, these methods are tailored to unidirectional text-to-motion synthesis and provide neither motion-to-text captioning nor a shared formulation that exploits the bidirectional dependencies between language and motion.

Unified Motion-Text Modeling. Beyond unidirectional synthesis, recent work studies unified models that handle motion generation and captioning within a shared framework [59, 53, 76, 62, 69, 50, 43, 5, 29, 63, 54]. TM2T [14] introduces motion tokens and trains separate translators for the two directions. MotionGPT [21] treats motion as a foreign language and unifies T2M and M2T under an autoregressive Transformer, while MotionGPT3 [75] extends this paradigm with a lightweight diffusion head to improve motion fidelity. However, the underlying causal backbone still imposes a fixed left-to-right generation order, which is misaligned with the inherently bidirectional dependencies between text and motion. Unimotion [25] jointly diffuses motion and frame-level text representations for motion synthesis and understanding. A concurrent work, DiMo [70], also adopts masked discrete diffusion for bidirectional motion-text modeling, using an RVQ predictor and reinforcement-learning fine-tuning to enhance motion fidelity and cross-modal alignment. BiMoGen instead focuses on the train-test misalignment and error accumulation of masked diffusion. By introducing generation-aware self-correction during both training and sampling, BiMoGen enables the model to revise unreliable predictions while retaining a simpler single-Transformer design over a single-layer VQ-VAE.

Masked Discrete Diffusion Models. Masked discrete diffusion generates discrete sequences by iteratively predicting masked tokens with a bidirectional Transformer, offering a nonautoregressive alternative to GPT style decoding [9, 19, 16, 11, 3, 52, 26, 60]. D3PM [2] formalizes diffusion over discrete state spaces, MaskGIT [4] introduces confidence guided parallel decoding for image synthesis, and SEDD [31] and LLaDA [35] extend related approaches to language modeling. GIDD [49] explores corruption with both masking and uniform noise alongside correction of generated tokens, while GASC retains mask only corruption and supervises predictions made from generated context. For bidirectional modeling of motion and text, the large difference in sequence length can bias unified prediction toward motion reconstruction, while errors in generated motion tokens can accumulate during iterative refinement. BiMoGen addresses these challenges through decoupled cross modal pretraining, conditional fine tuning, and generation-aware self correction.

3 Method

In this section, we present BiMoGen, a unified masked discrete diffusion framework for bidirectional motion-text modeling. We first formulate the unified bidirectional motion-text generation task (Sec. 3.1), then introduce the decoupled training procedure (Sec. 3.2), and finally present generation-aware self-correction for training and sampling (Sec. 3.3).

3.1 Unified Masked Discrete Diffusion Model

Discrete Representation. We employ a pretrained VQ-VAE [47, 21] as the motion tokenizer, whose encoder $\mathcal{E}$ discretizes a motion sequence $\mathbf{m} = (m_1, \ldots, m_F)$ of F frames into a sequence of $N_m = F/r$ motion tokens $\mathbf{x}^m = (x_1^m, \ldots, x_{N_m}^m)$, where r denotes the temporal downsampling rate, and whose decoder $\mathcal{D}$ maps motion tokens back to a continuous motion sequence at inference time. Let $\mathbf{x}^c = (x_1^c, \ldots, x_{N_c}^c)$ denote the corresponding text tokens of length N_c. The motion vocabulary is appended to the text vocabulary so that both modalities share a unified token space [21]. The two modalities are concatenated into a single sequence $\mathbf{x}_0 = [\mathbf{x}^c, \mathbf{x}^m]$ of length $N = N_c + N_m$, which serves as the unified input for masked discrete diffusion.

Forward Process and Training Objective. We follow the masked discrete diffusion formulation [35]. For a timestep t sampled uniformly from $(0, 1]$, each token in $\mathbf{x}_0$ is independently replaced by a mask token `[M]` with probability t, yielding a corrupted sequence $\mathbf{x}_t$. A bidirectional Transformer

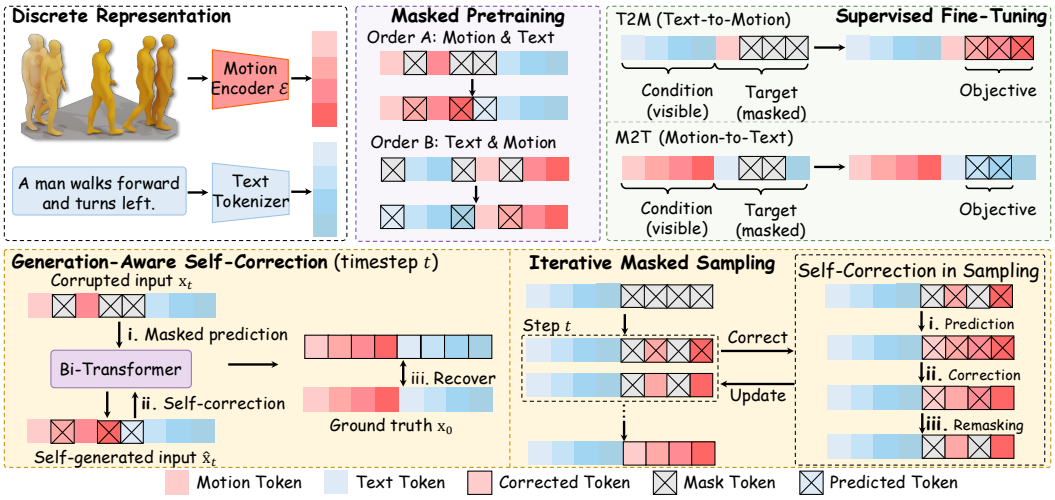

Figure 2: Overview of the BiMoGen framework. **Top:** motion and text are represented as discrete tokens in a unified vocabulary, and the model is trained in two-stages, masked pretraining over symmetrically corrupted motion-text sequences and supervised fine-tuning for T2M and M2T generation. **Bottom:** a self-correction mechanism is applied at both training and sampling, where the model takes its own predictions as input and learns to recover the ground-truth, and analogously revises committed tokens during iterative masked decoding.

$p_\theta(\cdot \mid \mathbf{x}_t)$ outputs a distribution over clean tokens at every position and is optimized by

$$\mathcal{L}_{\text{MDM}}(\theta) = -\mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_t}\left[\frac{1}{t}\sum_{i=1}^{N}\mathbf{1}[x_t^i = \texttt{[M]}]\log p_\theta(x_0^i \mid \mathbf{x}_t)\right], \tag{1}$$

where the indicator $\mathbf{1}[\cdot]$ restricts supervision to masked positions and the factor $1/t$ arises from the variational bound and reweights contributions across different mask ratios. The corresponding reverse process iteratively predicts masked tokens conditioned on currently visible ones, which we instantiate at sampling time in Sec. 3.3.

3.2 Decoupled Uni- and Cross-Modal Training

Bidirectional motion-text generation ultimately requires the model to produce one modality conditioned on the other, but conditional generation can only succeed once a reliable cross-modal correspondence has been established. Optimizing both objectives jointly forces the model to generate target tokens before alignment has emerged, which destabilizes training. We therefore introduce Decoupled Uni- and Cross-Modal Training (DUCMT): masked pretraining first learns motion-text correspondence over unified sequences, and supervised fine-tuning then specializes the model for the T2M and M2T directions.

Masked Pretraining for Motion-Text Correspondence. The first stage jointly models text and motion under the masked prediction objective in Eq. (1), with the loss denoted $\mathcal{L}_{\text{PT}}$. Each paired sequence is formed by concatenating $\mathbf{x}^c$ and $\mathbf{x}^m$ in either order with equal probability, which prevents the model from relying on a fixed modality order as a positional prior. Both modalities are corrupted by the same forward process, and the model is supervised on all masked positions regardless of modality. This symmetric corruption provides direct token-level supervision to both sides and encourages the Transformer to learn motion-text correspondence.

Supervised Fine-tuning for Bidirectional Generation. The second stage adapts the pretrained backbone to conditional generation by fine-tuning on pairs $(\mathbf{x}^{\text{cond}}, \mathbf{x}^{\text{tgt}})$. For T2M, the condition is text $\mathbf{x}^c$ and the target is the motion $\mathbf{x}^m$. For M2T, the assignment is reversed. The condition is kept visible while only the target is corrupted by the forward process, and the masked prediction objective

is computed only at masked target positions. The supervised fine-tuning loss is formulated as

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_t^{\mathrm{tgt}}} \left[\frac{1}{t} \sum_{i=1}^{N_{\mathrm{tgt}}} \mathbf{1}[x_t^{\mathrm{tgt},i} = \texttt{[M]}] \log p_\theta\Big(x_0^{\mathrm{tgt},i} \mid \mathbf{x}^{\mathrm{cond}}, \mathbf{x}_t^{\mathrm{tgt}} \Big) \right], \tag{2}$$

where N_{tgt} denotes the length of $\mathbf{x}^{\mathrm{tgt}}$. To support classifier-free guidance at sampling time, we further replace the entire condition with `[M]` tokens at a 10% dropout rate so that the model jointly learns the conditional and unconditional distributions [17].

Auxiliary Motion-only Supervision. Beyond the two cross-modal directions, we include a motion-to-motion (M2M) objective, where motion tokens at randomly sampled positions serve as the condition and the rest as the target. Text descriptions capture selected aspects of a motion sequence, whereas M2M provides dense intra-modal supervision to strengthen the motion prior of the unified model.

3.3 Generation-Aware Self-Correction

Both training stages above supervise p_θ on inputs whose visible positions hold ground-truth tokens, whereas at sampling time visible target positions are filled by tokens that p_θ predicted in earlier steps. This train-test misalignment between training on ground-truth context and inference on self-generated context is a core limitation of masked diffusion training. Once a token is unmasked during sampling, it remains as fixed context for all subsequent steps, and an error made under high mask ratios propagates rather than gets corrected. We address this mismatch with Generation-Aware Self-Correction (GASC), which trains the model to recover ground-truth tokens from its own predictions and further refines previously unmasked tokens during sampling.

Training with Self-Correction. Let $\mathcal{G}$ denote the generation region of the current stage, covering the entire sequence in pretraining and only the target in supervised fine-tuning. Given a corrupted input $\mathbf{x}_t$, the model first performs the standard masked prediction pass. We then construct a self-generated input $\tilde{\mathbf{x}}$ by replacing tokens in $\mathcal{G}$ with the model's detached argmax predictions, while tokens outside $\mathcal{G}$ remain unchanged

$$\tilde{x}^i = \mathrm{sg}\Big(\arg\max_v \, p_\theta(v \mid \mathbf{x}_t)_i \Big), \quad i \in \mathcal{G}, \tag{3}$$

where $\mathrm{sg}(\cdot)$ stops gradients through the discrete prediction. The same Transformer is then applied to $\tilde{\mathbf{x}}$ and trained to recover the ground-truth tokens over the entire generation region

$$\mathcal{L}_{\mathrm{SC}}(\theta) = -\mathbb{E}_{t,\mathbf{x}_0,\tilde{\mathbf{x}}} \left[\frac{1}{t} \sum_{i \in \mathcal{G}} \log p_\theta(x_0^i \mid \tilde{\mathbf{x}}) \right]. \tag{4}$$

Unlike the standard masked diffusion loss, this correction loss is applied to all positions in $\mathcal{G}$, since committed tokens during sampling may also be incorrect and should remain revisable. The full objective for each training stage is

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{stage}}(\theta) + \lambda \, \mathcal{L}_{\mathrm{SC}}(\theta), \quad \mathrm{stage} \in \{\mathrm{PT}, \mathrm{SFT}\}, \tag{5}$$

where λ controls the strength of self-correction supervision.

Sampling with Self-Correction. At inference time, BiMoGen generates the target sequence by iteratively refining a fully masked target. Given a condition $\mathbf{x}^{\mathrm{cond}}$, we initialize $\mathbf{x}^{\mathrm{tgt}}$ as a sequence of `[M]` tokens. For T2M, the target motion length is given. For M2T, we generate up to a maximum text length and truncate the output at the first `[EOS]` token.

Sampling proceeds for T uniformly spaced timesteps. At each timestep, p_θ predicts a distribution over clean tokens at all masked positions. We compute the confidence of each masked position as the maximum predicted probability and progressively unmask the most confident positions, while leaving the remaining positions masked for subsequent steps. We further apply classifier-free guidance [17, 6] during sampling, where the conditional and unconditional predictions are linearly combined as

$$p_\theta^w\big(\cdot \mid \mathbf{x}^{\mathrm{cond}}, \mathbf{x}^{\mathrm{tgt}}\big) = w \, p_\theta\big(\cdot \mid \mathbf{x}^{\mathrm{cond}}, \mathbf{x}^{\mathrm{tgt}}\big) + (1-w) \, p_\theta\big(\cdot \mid \emptyset, \mathbf{x}^{\mathrm{tgt}}\big), \tag{6}$$

where w is the guidance scale and $w > 1$ strengthens the conditioning effect. The unconditional case $\emptyset$ corresponds to replacing the condition with `[M]`, matching the dropout used during fine-tuning.

Table 1: **Comparison of bidirectional motion-text generation performance on HumanML3D.** Best/second-best unified results are **bold**/underlined.

Method	T2M						M2T						
	R@1↑	R@2↑	R@3↑	FID↓	Div→	MM Dist↓	R@1↑	R@3↑	BLEU@1↑	BLEU@4↑	ROUGE-L↑	CIDEr↑	BERTScore↑
Real	$0.511^{\pm0.003}$	$0.703^{\pm0.003}$	$0.797^{\pm0.002}$	$0.002^{\pm0.000}$	$9.503^{\pm0.065}$	$2.974^{\pm0.008}$	0.523	0.828	–	–	–	–	–
T2M-Only Models													
MDM [46]	–	–	$0.611^{\pm0.007}$	$0.544^{\pm0.044}$	$9.559^{\pm0.086}$	$5.566^{\pm0.027}$	–	–	–	–	–	–	–
MotionDiffuse [68]	$0.491^{\pm0.001}$	$0.681^{\pm0.001}$	$0.782^{\pm0.001}$	$0.630^{\pm0.001}$	$9.410^{\pm0.049}$	$3.113^{\pm0.001}$	–	–	–	–	–	–	–
MLD [6]	$0.481^{\pm0.003}$	$0.673^{\pm0.003}$	$0.772^{\pm0.002}$	$0.473^{\pm0.013}$	$9.724^{\pm0.082}$	$3.196^{\pm0.010}$	–	–	–	–	–	–	–
MoMask [15]	$0.521^{\pm0.002}$	$0.713^{\pm0.002}$	$0.807^{\pm0.002}$	$0.045^{\pm0.002}$	9.620	$2.958^{\pm0.008}$	–	–	–	–	–	–	–
T2M-GPT [65]	$0.492^{\pm0.003}$	$0.679^{\pm0.002}$	$0.775^{\pm0.002}$	$0.141^{\pm0.005}$	$9.722^{\pm0.082}$	$3.121^{\pm0.009}$	–	–	–	–	–	–	–
ReMoDiffuse [67]	$0.510^{\pm0.005}$	$0.698^{\pm0.006}$	$0.795^{\pm0.004}$	$0.103^{\pm0.004}$	$9.018^{\pm0.075}$	$2.974^{\pm0.016}$	–	–	–	–	–	–	–
MoGenTS [64]	$0.529^{\pm0.003}$	$0.719^{\pm0.002}$	$0.812^{\pm0.002}$	$0.033^{\pm0.001}$	$9.570^{\pm0.077}$	$2.867^{\pm0.006}$	–	–	–	–	–	–	–
MotionLCM [7]	$0.502^{\pm0.003}$	$0.698^{\pm0.002}$	$0.798^{\pm0.002}$	$0.304^{\pm0.012}$	$9.607^{\pm0.066}$	$3.012^{\pm0.007}$	–	–	–	–	–	–	–
BAMM [37]	$0.525^{\pm0.002}$	–	$0.814^{\pm0.003}$	$0.055^{\pm0.002}$	–	2.919	–	–	–	–	–	–	–
BiPO [18]	$0.523^{\pm0.003}$	–	$0.809^{\pm0.002}$	$0.030^{\pm0.002}$	–	$2.880^{\pm0.009}$	–	–	–	–	–	–	–
CoMo [20]	$0.502^{\pm0.002}$	$0.692^{\pm0.007}$	$0.790^{\pm0.002}$	$0.262^{\pm0.004}$	$9.936^{\pm0.066}$	$3.032^{\pm0.015}$	–	–	–	–	–	–	–
MotionMamba [73]	$0.502^{\pm0.003}$	$0.693^{\pm0.002}$	$0.792^{\pm0.002}$	$0.281^{\pm0.009}$	$9.871^{\pm0.084}$	$3.060^{\pm0.058}$	–	–	–	–	–	–	–
EnergyMoGen [66]	$0.526^{\pm0.003}$	$0.718^{\pm0.003}$	$0.815^{\pm0.002}$	$0.176^{\pm0.006}$	$9.500^{\pm0.091}$	$2.931^{\pm0.007}$	–	–	–	–	–	–	–
Separated Bidirectional Models													
TM2T [14]	$0.424^{\pm0.003}$	$0.618^{\pm0.003}$	$0.729^{\pm0.002}$	$1.501^{\pm0.017}$	$8.589^{\pm0.076}$	$3.467^{\pm0.011}$	0.516	0.823	48.9	7.0	38.1	16.8	32.2
LaMP [27]	$0.557^{\pm0.003}$	$0.751^{\pm0.002}$	$0.843^{\pm0.001}$	$0.032^{\pm0.002}$	$9.571^{\pm0.069}$	$2.759^{\pm0.007}$	0.547	0.831	47.8	13.0	37.1	28.9	–
MG-Mo.LLM [59]	$0.516^{\pm0.002}$	$0.706^{\pm0.002}$	$0.802^{\pm0.003}$	$0.303^{\pm0.010}$	$9.960^{\pm0.073}$	$2.952^{\pm0.009}$	0.592	0.866	–	8.1	–	–	36.7
Unified Bidirectional Models													
MotionGPT [21]	$0.492^{\pm0.003}$	$0.681^{\pm0.003}$	$0.778^{\pm0.002}$	$0.232^{\pm0.008}$	$9.528^{\pm0.071}$	$3.096^{\pm0.008}$	0.543	0.827	48.2	12.5	37.4	29.2	32.4
MotionGPT2 [53]	$0.496^{\pm0.002}$	$0.691^{\pm0.003}$	$0.782^{\pm0.004}$	$0.191^{\pm0.004}$	$9.860^{\pm0.026}$	$3.080^{\pm0.013}$	0.558	0.838	48.7	13.8	37.6	29.8	32.6
MotionGPT3 [75]	$0.553^{\pm0.003}$	$\mathbf{0.747}^{\pm0.002}$	$0.837^{\pm0.003}$	$0.208^{\pm0.006}$	$9.700^{\pm0.096}$	$\mathbf{2.725}^{\pm0.008}$	0.573	0.864	59.1	19.4	46.2	28.7	35.2
MoTe [61]	$0.548^{\pm0.002}$	$0.737^{\pm0.002}$	$0.825^{\pm0.002}$	$0.075^{\pm0.004}$	–	$2.867^{\pm0.012}$	**0.577**	**0.871**	46.7	11.2	37.4	31.5	30.3
DiMo [70]	0.528	0.724	0.818	**0.047**	9.419	2.862	**0.577**	0.855	**64.2**	**22.7**	**47.1**	58.1	37.7
BiMoGen (Ours)	$\mathbf{0.555}^{\pm0.003}$	$0.744^{\pm0.003}$	$\mathbf{0.841}^{\pm0.002}$	$0.069^{\pm0.007}$	$\mathbf{9.524}^{\pm0.079}$	$2.733^{\pm0.008}$	0.573	0.866	60.1	20.1	44.4	**60.2**	**38.5**

Table 2: **Comparison of bidirectional motion-text generation performance on KIT-ML.** We report Text-to-Motion (T2M) and Motion-to-Text (M2T) results across two method categories. The best results within unified models are highlighted in **bold**, and the second-best are underlined.

Method	T2M						M2T						
	R@1↑	R@2↑	R@3↑	FID↓	Div→	MM Dist↓	R@1↑	R@3↑	BLEU@1↑	BLEU@4↑	ROUGE-L↑	CIDEr↑	BERTScore↑
Real	$0.424^{\pm0.005}$	$0.649^{\pm0.006}$	$0.779^{\pm0.006}$	$0.031^{\pm0.004}$	$11.080^{\pm0.097}$	$2.788^{\pm0.012}$	0.399	0.793	–	–	–	–	–
Separated Bidirectional Models													
TM2T [14]	$0.280^{\pm0.005}$	$0.463^{\pm0.006}$	$0.587^{\pm0.005}$	$3.599^{\pm0.153}$	$9.473^{\pm0.117}$	$4.591^{\pm0.026}$	0.359	0.668	46.7	18.4	44.2	79.5	23.0
LaMP [27]	$0.479^{\pm0.006}$	$0.691^{\pm0.005}$	$0.826^{\pm0.005}$	$0.141^{\pm0.013}$	$10.929^{\pm0.101}$	$2.704^{\pm0.018}$	0.540	0.844	–	–	–	–	–
Unified Bidirectional Models													
MotionGPT [21]	$0.366^{\pm0.005}$	$0.558^{\pm0.004}$	$0.680^{\pm0.005}$	$0.510^{\pm0.016}$	$10.350^{\pm0.084}$	$3.527^{\pm0.021}$	–	–	–	–	–	–	–
MotionGPT2 [53]	$\mathbf{0.427}^{\pm0.003}$	$0.627^{\pm0.002}$	$\mathbf{0.764}^{\pm0.003}$	$0.614^{\pm0.005}$	$\mathbf{11.256}^{\pm0.026}$	$3.164^{\pm0.013}$	–	–	–	–	–	–	–
MoTe [61]	$0.419^{\pm0.005}$	$0.627^{\pm0.005}$	$0.741^{\pm0.004}$	$0.256^{\pm0.018}$	–	$3.216^{\pm0.036}$	**0.421**	**0.765**	44.9	14.1	41.8	55.6	35.9
DiMo [70]	0.406	0.620	0.741	**0.206**	10.892	2.983	0.396	0.723	**52.5**	**17.8**	**48.0**	**68.7**	37.7
BiMoGen (Ours)	$0.421^{\pm0.004}$	$\mathbf{0.632}^{\pm0.007}$	$0.733^{\pm0.006}$	$0.222^{\pm0.019}$	$10.880^{\pm0.096}$	$\mathbf{2.763}^{\pm0.022}$	0.405	0.734	49.8	17.3	46.1	62.2	**40.4**

Iterative sampling uses unmasked tokens as context for subsequent steps, so errors made under heavily masked states can affect later predictions. We address this by invoking self-correction at a set of selected sampling steps $\mathcal{R} \subseteq \{1, \ldots, T\}$. When triggered, the current partially masked target is first predicted into a complete target candidate $\tilde{\mathbf{x}}^{\mathrm{tgt}}$, which is then fed back to the model for correction. The corrected prediction is written back only to positions that have already been unmasked

$$x^{\mathrm{tgt},i} \leftarrow \arg\max_{v} p_\theta\big(v \mid \mathbf{x}^{\mathrm{cond}}, \tilde{\mathbf{x}}^{\mathrm{tgt}}\big)_i, \quad \text{for } x^{\mathrm{tgt},i} \neq \texttt{[M]}. \tag{7}$$

The remaining masked positions are kept as `[M]` and deferred to subsequent sampling steps. This correction step allows tokens committed in earlier iterations to be revised as the context becomes clearer without prematurely filling unresolved positions. After iterative decoding, the predicted motion tokens are decoded back to a continuous motion sequence by the VQ-VAE decoder $\mathcal{D}$. The full sampling procedure is summarized in Appendix C.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate BiMoGen mainly on HumanML3D [13], a standard benchmark for bidirectional text and motion generation. HumanML3D contains 14,616 motion sequences from

Table 3: Ablation on the training strategy on HumanML3D. PT denotes masked pretraining for cross-modal correspondence, SFT denotes supervised fine-tuning for bidirectional generation, and SC denotes the self-correction objective. **Bold** denotes the best performance.

Training Strategy			T2M						M2T						
PT	SFT	SC	R@1 ↑	R@2 ↑	R@3 ↑	FID ↓	Div →	MM Dist ↓	R@1 ↑	R@3 ↑	BLEU@1 ↑	BLEU@4 ↑	ROUGE-L ↑	CIDEr ↑	BERTScore ↑
Real			0.511	0.703	0.797	0.002	9.503	2.974	0.523	0.828	–	–	–	–	–
✓	✗	✗	0.416	0.604	0.730	1.219	10.189	3.650	0.434	0.736	24.3	5.5	26.4	6.7	16.8
✗	✓	✗	0.466	0.633	0.726	0.325	9.546	3.259	0.474	0.779	54.3	16.7	34.9	41.3	26.4
✓	✓	✗	0.514	0.697	0.789	0.102	9.616	3.017	0.547	0.832	56.1	16.4	42.9	44.0	36.7
✓	✓	✓	**0.555**	**0.744**	**0.841**	**0.069**	**9.524**	**2.733**	**0.573**	**0.866**	**60.1**	**20.1**	**44.4**	**60.2**	**38.5**

Table 4: Ablation of the trade-off between quality and computational cost. Representative bidirectional methods are included for reference under the same evaluation protocol. Lat. denotes Latency.

Method	Text-to-Motion					Motion-to-Text						Computational Cost	
	R@1↑	R@3↑	FID↓	MM Dist↓	Lat. (s)↓	R@1↑	R@3↑	BLEU@1↑	BLEU@4↑	ROUGE-L↑	BERTScore↑	#Params	FLOPs
MotionGPT [21]	0.492	0.778	0.232	3.096	1.04	0.543	0.827	48.2	12.5	37.4	32.4	220M	7.45T
MotionGPT3 [75]	0.553	0.837	0.208	**2.725**	1.02	0.573	0.864	59.1	19.4	46.2	35.2	238M	11.00T
MG-Mo.LLM [59]	0.516	0.802	0.303	2.952	1.09	**0.592**	**0.866**	–	8.1	–	36.7	220M	1.66T
DiMo (20 steps) [70]	0.528	0.818	**0.050**	2.862	1.55	0.568	0.845	**62.5**	**22.0**	**47.3**	35.4	473M	2.56T
Ours (5 steps)	0.533	0.832	0.118	2.787	**0.20**	0.467	0.749	57.6	16.8	44.3	28.9	334M	0.34T
Ours (10 steps)	0.543	0.840	0.082	2.774	0.35	0.525	0.823	59.1	17.3	44.8	33.3	334M	0.59T
Ours (20 steps)	**0.555**	**0.841**	0.069	2.733	0.67	0.573	**0.866**	60.1	20.1	44.4	38.5	334M	1.08T
Ours (30 steps)	0.550	0.840	0.071	2.756	0.95	0.573	0.853	59.2	19.0	46.0	**41.2**	334M	1.57T

AMASS [32] and HumanAct12 [12], paired with 44,970 textual descriptions. We also report results on KIT-ML [38], a smaller benchmark with 3,911 motion sequences and 6,278 textual annotations. For both datasets, we follow the settings and motion representations adopted in prior work [21, 6].

Evaluation Protocol. Following the evaluation protocol of Guo et al. [13], we use the pretrained motion-text feature extractor and average results over 20 evaluation runs. For text-to-motion (T2M) we report Top-k retrieval accuracy (R@k), Fréchet Inception Distance (FID), Diversity, and MM Dist, the average feature-space distance between paired text and generated motion. For motion-to-text (M2T) we report retrieval accuracy R@1/R@3 against ground-truth motions, together with BLEU, ROUGE-L, CIDEr, and BERTScore for caption quality [48, 28, 72, 36]. See Appendix B.1 for details.

Implementation Details. The bidirectional Transformer follows the LLaDA backbone [35] with 20 layers and a hidden size of 1024. Motion tokens come from the single-layer VQ-VAE used in MotionGPT [21, 47], with a temporal downsampling rate of 4. We pretrain for 100 epochs at learning rate 2e-4 and fine-tune for 200 epochs at 8e-5, both with AdamW [30]. Fine-tuning batches mix T2M, M2T, and the auxiliary M2M objective at an 8:1:1 ratio, with 10% condition dropout for CFG [17]. The self-correction weight λ is set to 1. At inference we use 20 sampling steps, CFG scale for T2M set to 4, and self-correction at $\mathcal{R}=\{T/4, T/2\}$. We train and evaluate on a single NVIDIA H100.

4.2 Main Results

Comparisons on Bidirectional Motion-Text Generation. Table 1 reports the main results on HumanML3D. Because BiMoGen is designed as a unified model for both T2M and M2T, we focus our comparisons on the unified bidirectional setting, while using T2M-Only and separated bidirectional methods only as references. On T2M, BiMoGen achieves the best retrieval performance among unified models, with an R@1 of 0.555 and an R@3 of 0.841. It also maintains competitive motion quality, with an FID of 0.069 and an MM Dist of 2.733. These results show that masked bidirectional prediction improves motion-text alignment over autoregressive unified models such as MotionGPT and MotionGPT3 [21, 75], while still producing realistic motions. DiMo [70] achieves lower FID, but differences in tokenization, architecture, and training preclude attributing this gap solely to tokenization. On M2T, BiMoGen achieves the highest CIDEr and BERTScore among unified models, reaching 60.2 and 38.5, respectively. This indicates a stronger ability in motion understanding, especially compared with autoregressive unified methods. Appendix B.2.1 compares unified and T2M-only training.

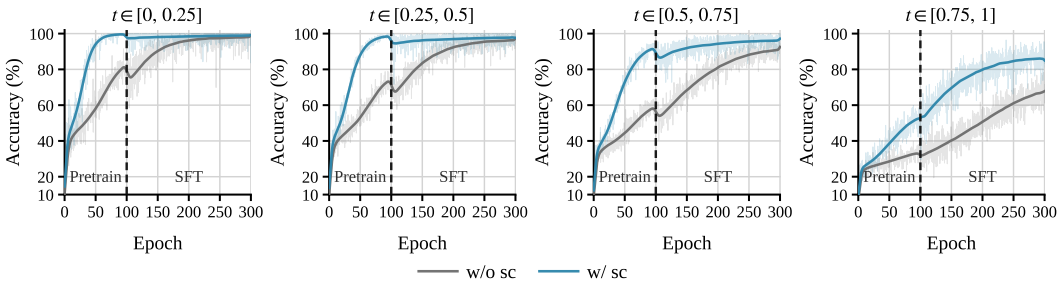

Figure 3: Training accuracy under different masking levels t. Each panel shows one masking interval. The dashed vertical line separates masked pretraining and supervised fine-tuning.

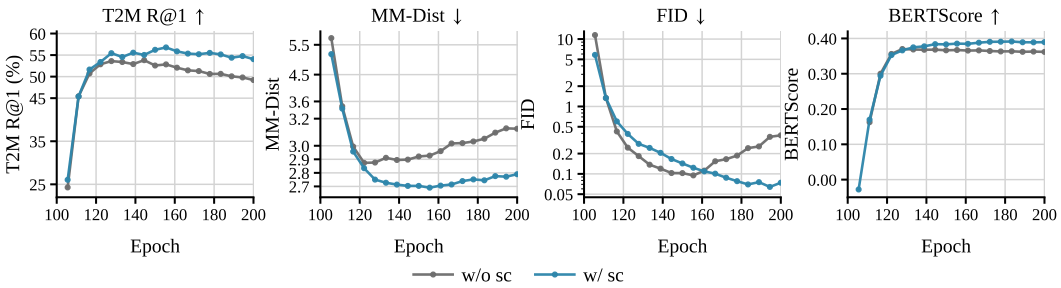

Figure 4: SFT evaluation over epochs. The first three panels correspond to T2M, and the last panel corresponds to M2T. Curves compare training with and without self-correction.

4.3 Ablation and Analysis

Effect of Training Strategy. Table 3 isolates the contribution of each training stage. PT alone learns useful cross-modal correspondence, but its high FID and weak captioning scores show that correspondence pretraining is not sufficient for bidirectional generation. SFT is therefore necessary for producing valid target sequences. Adding PT before SFT improves alignment and motion quality. MM Dist decreases from 3.259 to 3.017, FID decreases from 0.325 to 0.102, and M2T R@1 increases from 0.474 to 0.547. These results support correspondence pretraining as an initialization for conditional generation.

Training and Sampling Self-Correction. Table 5 separates the two components. With sampling SC disabled, training SC raises R@1 from 0.514 to 0.551, lowers FID from 0.102 to 0.070, and improves CIDEr from 44.0 to 58.8. Sampling SC alone degrades these metrics. After training with SC, sampling correction further raises R@1 to 0.555 and CIDEr to 60.2. The main gain therefore comes from training on self-generated context, which enables further refinement at sampling time. Figure 3 shows the same effect in the training dynamics. At low mask ratios, SC mainly accelerates convergence and reaches a similar final accuracy. As the mask ratio increases, the advantage becomes larger and persists throughout training, especially at $t \in [0.75, 1]$. This regime matches the early stage of iterative sampling, where the model predicts from sparse context and then conditions on its previous outputs. Training with SC therefore reduces the mismatch between training on corrupted ground-truth context and inference on self-generated context, leading to more stable generation in both directions. Figure 4 further reports the SFT evaluation over epochs on both T2M and M2T, comparing training with and without SC.

Sampling Correction Schedule. Table 6 studies the timing of SC during masked sampling. Compared with *Disabled*, applying SC at both $T/4$ and $T/2$ improves T2M alignment, increasing R@1 and R@3 to 0.555 and 0.841 and reducing MM Dist from 2.745 to 2.733, while maintaining a competitive FID of 0.069. It also improves M2T R@1, BLEU@4, and CIDEr over *Disabled*, with a moderate latency increase from 0.55s to 0.67s. Single-step correction at $T/4$ or $T/2$ alone brings less consistent gains, while late correction at $3T/4$ achieves the lowest FID but weakens text-motion alignment, suggesting that late-stage revision mainly adjusts motion statistics rather than improving

Table 5: Training and sampling self-correction on HumanML3D. All variants use PT and SFT. R@1, FID, and MM Dist evaluate T2M; CIDEr evaluates M2T. Sampling SC uses $\mathcal{R} = \{T/4, T/2\}$.

Training SC	Sampling SC	R@1 ↑	FID ↓	MM Dist ↓	CIDEr ↑
✗	✗	0.514	0.102	3.017	44.0
✗	✓	0.503	0.118	3.132	42.9
✓	✗	0.551	0.070	2.745	58.8
✓	✓	0.555	0.069	2.733	60.2

Table 6: Ablation on the SC schedule during sampling on HumanML3D. *Disabled* removes SC during sampling only. All variants use the same number of sampling steps T. For T2M, the CFG scale w is fixed to its default value. CFG is not used for M2T. Lat. denotes Latency.

SC Schedule $\mathcal{R}$	T2M						M2T						
	R@1↑	R@3↑	FID↓	Div→	MM Dist↓	Lat. (s)↓	R@1↑	R@3↑	BLEU@1↑	BLEU@4↑	ROUGE-L↑	CIDEr↑	BERTScore↑
Disabled	0.551	0.838	0.070	9.870	2.745	**0.55**	0.568	0.864	59.2	18.9	45.6	58.8	39.6
$T/4$	0.553	0.836	0.067	9.397	2.764	0.61	0.570	0.866	59.2	19.0	45.6	58.8	38.6
$T/2$	0.551	0.837	0.070	9.697	2.782	0.61	0.570	0.866	59.2	18.9	45.6	58.9	38.6
$3T/4$	0.542	0.831	**0.056**	9.423	2.833	0.61	0.570	0.866	59.2	18.9	45.6	59.6	38.8
Every step	0.529	0.814	0.066	9.570	2.948	0.78	**0.580**	**0.868**	**60.2**	19.7	**45.8**	**61.2**	**41.0**
$T/4$, $T/2$	**0.555**	**0.841**	0.069	**9.524**	**2.733**	0.67	0.573	0.866	60.1	**20.1**	44.4	60.2	38.5

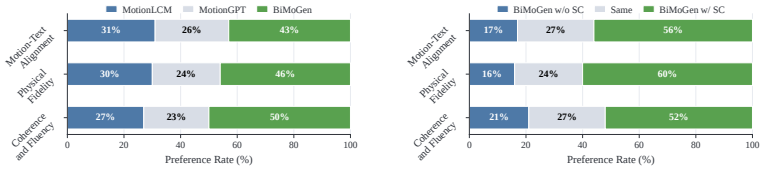

Figure 5: **User study.** Preference rates on 30 HumanML3D test prompts. Left: comparison with MotionLCM and MotionGPT. Right: ablation of SC with an additional "Same" option.

cross-modal consistency. Applying SC at every step further improves a few captioning metrics but degrades T2M retrieval. We therefore use $\mathcal{R} = \{T/4, T/2\}$ as the default.

Table 7 further analyzes the correction behavior. The modification rate decreases from 31.82% at $T/4$ to 22.38% at T, indicating more frequent revisions at earlier sampling steps. The stability of revised tokens remains high between 82.73% and 84.70%, so most revisions agree with the next step prediction. Corrections are more frequent at earlier sampling steps, and a larger fraction of the revised tokens move toward ground truth, suggesting that early correction is more effective at reducing prediction errors.

Computation-Quality Trade-Off. Table 4 analyzes the effect of sampling steps and compares the computational cost with representative baselines. Increasing the number of steps from 5 to 20 improves retrieval in both generation directions. For T2M, FID decreases from 0.118 to 0.069. For M2T, retrieval accuracy also increases substantially, with BERTScore increasing from 28.9 to 38.5. The setting with 20 steps gives the best overall balance. It attains the strongest T2M retrieval and the best R@3 among the compared bidirectional methods, while using 1.08T FLOPs. This is substantially lower than MotionGPT and DiMo, which require 7.45T FLOPs and 2.56T FLOPs. Since 30 steps brings no consistent gain, we use 20 denoising steps in the main experiments.

Effect of Motion Tokenizer. We study the effect of motion tokenizer by replacing the whole-body VQ-VAE with the Part VQ-VAE from ParCo [80], which quantizes different body parts with independent codebooks. As shown in Table 8, this leads to worse performance. We attribute the drop to the increased learning difficulty introduced by part-wise tokenization under masked bidirectional generation. Part VQ-VAE produces multiple fine-grained tokens, and modeling their dependencies with full attention requires the model to learn both temporal dynamics and cross-part coordination simultaneously, which may be challenging under our current model capacity. The flattened RVQ variant similarly introduces a longer sequence with dependencies across residual levels and yields

Table 7: Self-correction behavior across sampling steps. *Modification* is the ratio of committed tokens revised by SC. *Pred. Stability* is the ratio of revised tokens that remain unchanged at the next step. Toward-GT and Away-from-GT report the direction of changes among revised tokens.

SC Step	Modification (%)	Pred. Stability (%)	Toward-GT (%)	Away-from-GT (%)
$T/4$	31.82	84.70	68.5	31.5
$T/2$	29.48	83.72	64.7	35.3
$3T/4$	25.72	83.74	60.2	39.8
T	22.38	82.73	55.8	44.2

Table 8: T2M results with different motion tokenizers on HumanML3D. Part VQ-VAE follows ParCo [80], which discretizes motion by body parts. VQ-VAE denotes the whole-body tokenizer [21]. * denotes base and residual RVQ codes flattened into one sequence.

Motion Tokenizer	Text-to-Motion Generation					
	R@1↑	R@2↑	R@3↑	FID↓	Div→	MM Dist↓
Real	0.511	0.703	0.797	0.002	9.503	2.974
Part VQ-VAE [80]	0.512	0.703	0.801	0.124	9.886	3.084
RVQ-VAE [15]*	0.484	0.665	0.766	0.487	9.992	3.114
VQ-VAE [21]	**0.555**	**0.744**	**0.841**	**0.069**	**9.524**	**2.733**

an FID of 0.487 under the same single-Transformer design. Together, these comparisons support whole-body tokenization for the evaluated masked generation framework.

User Study. As shown in Figure 5, BiMoGen achieves the highest preference rates across all three criteria. Adding SC further improves user preference, indicating better text alignment, physical plausibility, and temporal consistency. See Appendix B.3 for the study protocol.

5 Conclusion

We presented BiMoGen, a unified masked discrete diffusion framework for bidirectional motion-text modeling. By formulating both T2M and M2T as iterative masked prediction in a shared motion-text token space, BiMoGen avoids the rigid left-to-right order of autoregressive decoding and enables bidirectional conditioning between language and motion. DUCMT establishes cross-modal correspondence before conditional fine-tuning, while GASC trains recovery from self-generated context and enables further sampling refinement. Experiments on HumanML3D and KIT-ML show consistent gains in both directions, with competitive performance among unified models.

Limitations. Our method employs a VQ-VAE motion tokenizer to convert continuous human motion into discrete tokens. As a result, generation quality is still partially bounded by the tokenizer's reconstruction fidelity and codebook expressiveness, especially for subtle or highly detailed motions. Moreover, the current formulation models motion at the sequence level and lacks explicit fine-grained control over individual body parts, such as hands, arms, or legs. Incorporating stronger part-aware tokenizers and more controllable representations is a promising direction for future work.

Acknowledgements

This work was supported by National Science Foundation of China (62672109, 62302093, 52441503), Jiangsu Province Natural Science Fund (BK20230833) and the Open Research Fund of the State Key Laboratory of Multimodal Artificial Intelligence Systems under Grant E5SP060116. We thank the Big Data Computing Center of Southeast University for providing the facility support on the numerical calculations.

References

[1] M. Arriola, A. Gokaslan, J. T. Chiu, Z. Yang, Z. Qi, J. Han, S. S. Sahoo, and V. Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*, 2025.

[2] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

[3] J. Bruce, M. D. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, E. Hughes, M. Lai, A. Mavalankar, R. Steigerwald, C. Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

[4] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11315–11325, 2022.

[5] C. Chen, J. Zhang, S. K. Lakshmikanth, Y. Fang, R. Shao, G. Wetzstein, L. Fei-Fei, and E. Adeli. The language of motion: Unifying verbal and non-verbal language of 3d human motion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6200–6211, 2025.

[6] X. Chen, B. Jiang, W. Liu, Z. Huang, B. Fu, T. Chen, and G. Yu. Executing your commands via motion diffusion in latent space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18000–18010, 2023.

[7] W. Dai, L.-H. Chen, J. Wang, J. Liu, B. Dai, and Y. Tang. Motionlcm: Real-time controllable motion generation via latent consistency model. In *ECCV*, pages 390–408, 2025.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[9] S. Dieleman, L. Sartran, A. Roshannai, N. Savinov, Y. Ganin, P. H. Richemond, A. Doucet, R. Strudel, C. Dyer, C. Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.

[10] K. Fan, S. Lu, M. Dai, R. Yu, L. Xiao, Z. Dou, J. Dong, L. Ma, and J. Wang. Go to zero: Towards zero-shot motion generation with million-scale data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13336–13348, 2025.

[11] S. Gong, S. Agarwal, Y. Zhang, J. Ye, L. Zheng, M. Li, C. An, P. Zhao, W. Bi, J. Han, et al. Scaling diffusion language models via adaptation from autoregressive models. *arXiv preprint arXiv:2410.17891*, 2024.

[12] C. Guo, X. Zuo, S. Wang, S. Zou, Q. Sun, A. Deng, M. Gong, and L. Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM international conference on multimedia*, pages 2021–2029, 2020.

[13] C. Guo, S. Zou, X. Zuo, S. Wang, W. Ji, X. Li, and L. Cheng. Generating diverse and natural 3d human motions from text. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[14] C. Guo, X. Zuo, S. Wang, and L. Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *European Conference on Computer Vision*, pages 580–597. Springer, 2022.

[15] C. Guo, Y. Mu, M. G. Javed, S. Wang, and L. Cheng. Momask: Generative masked modeling of 3d human motions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1900–1910, 2024.

[16] Z. He, T. Sun, Q. Tang, K. Wang, X.-J. Huang, and X. Qiu. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 4521–4534, 2023.

[17] J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[18] S.-E. Hong, S. Lim, J. Hwang, M. Chang, and H. Kang. Bipo: Bidirectional partial occlusion network for text-to-motion synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 32–42, March 2026.

[19] M. Hu, C. Zheng, Z. Yang, T.-J. Cham, H. Zheng, C. Wang, D. Tao, and P. N. Suganthan. Unified discrete diffusion for simultaneous vision-language generation. In *The Eleventh International Conference on Learning Representations*.

[20] Y. Huang, W. Wan, Y. Yang, C. Callison-Burch, M. Yatskar, and L. Liu. Como: Controllable motion generation through language guided pose code editing. In *European Conference on Computer Vision*, page 180–196. Springer-Verlag, 2024. ISBN 978-3-031-73396-3.

[21] B. Jiang, X. Chen, W. Liu, J. Yu, G. Yu, and T. Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36:20067–20079, 2023.

[22] H. Kong, K. Gong, D. Lian, M. B. Mi, and X. Wang. Priority-centric human motion generation in discrete latent space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14806–14816, 2023.

[23] J. Kuang, H. Wang, C. Han, Y. Zhang, and J. Gui. Zero-shot skeleton-based action recognition with dual visual-text alignment. *Pattern Recognition*, page 112342, 2025.

[24] J. Kuang, H. Wang, and J. Gui. Genprior: Unleashing text-to-motion generative priors for zero-shot skeleton-based action recognition. *arXiv preprint arXiv:2608.02236*, 2026.

[25] C. Li, J. Chibane, Y. He, N. Pearl, A. Geiger, and G. Pons-Moll. Unimotion: Unifying 3d human motion synthesis and understanding. *arXiv preprint arXiv:2409.15904*, 2024.

[26] S. Li, K. Kallidromitis, H. Bansal, A. Gokul, Y. Kato, K. Kozuka, J. Kuen, Z. Lin, K.-W. Chang, and A. Grover. Lavida: A large diffusion language model for multimodal understanding. *arXiv preprint arXiv:2505.16839*, 2025.

[27] Z. Li, W. Yuan, Y. HE, L. Qiu, S. Zhu, X. Gu, W. Shen, Y. Dong, Z. Dong, and L. T. Yang. LaMP: Language-motion pretraining for motion generation, retrieval, and captioning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[28] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics, 2004.

[29] Z. Ling, B. Han, S. Li, J. Cheng, H. Shen, and C. Zou. Versatilemotion: A unified framework for motion synthesis and comprehension. *arXiv preprint arXiv:2411.17335*, 2024.

[30] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[31] A. Lou, C. Meng, and S. Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

[32] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5442–5451, 2019.

[33] Z. Meng, Z. Han, X. Peng, Y. Xie, and H. Jiang. Absolute coordinates make motion generation easy. *arXiv preprint arXiv:2505.19377*, 2025.

[34] Z. Meng, Y. Xie, X. Peng, Z. Han, and H. Jiang. Rethinking diffusion for text-driven human motion generation: Redundant representations, evaluation, and masked autoregression. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27859–27871. IEEE, 2025.

[35] S. Nie, F. Zhu, Z. You, X. Zhang, J. Ou, J. Hu, J. Zhou, Y. Lin, J.-R. Wen, and C. Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

[36] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[37] E. Pinyoanuntapong, M. U. Saleem, P. Wang, M. Lee, S. Das, and C. Chen. Bamm: bidirectional autoregressive motion model. In *European Conference on Computer Vision*, pages 172–190. Springer, 2024.

[38] M. Plappert, C. Mandery, and T. Asfour. The kit motion-language dataset. *Big data*, 4(4): 236–252, 2016.

[39] P. Ruiz-Ponce, G. Barquero, C. Palmero, S. Escalera, and J. García-Rodríguez. Mixermdm: Learnable composition of human motion diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12380–12390, 2025.

[40] S. S. Sahoo, M. Arriola, Y. Schiff, A. Gokaslan, E. Marroquin, J. T. Chiu, A. Rush, and V. Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

[41] X. Tan, W. Weng, H. Wang, F. Zhao, X. Geng, and L. Wang. Motionrft: Unified reinforcement fine-tuning for text-to-motion generation. *arXiv preprint arXiv:2603.27185*.

[42] X. Tan, H. Wang, X. Geng, and P. Zhou. Sopo: Text-to-motion generation using semi-online preference optimization. *Advances in Neural Information Processing Systems*, 2025.

[43] X. Tan, J. Liu, B.-B. Gao, Y. Fan, X. Jiang, C. Wang, H. Wang, and F. Zheng. When policy entropy constraint fails: Preserving diversity in flow-based rlhf via perceptual entropy. 2026.

[44] X. Tan, W. Weng, H. Lei, and H. Wang. Easytune: Efficient step-aware fine-tuning for diffusion-based motion generation. In *International Conference on Learning Representations*, 2026.

[45] G. Tevet, B. Gordon, A. Hertz, A. H. Bermano, and D. Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *European Conference on Computer Vision*, pages 358–374. Springer, 2022.

[46] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-or, and A. H. Bermano. Human motion diffusion model. In *International Conference on Learning Representations*, 2023.

[47] A. van den Oord, O. Vinyals, and k. kavukcuoglu. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[48] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[49] D. von Rütte, J. Fluri, Y. Ding, A. Orvieto, B. Schölkopf, and T. Hofmann. Generalized interpolating discrete diffusion. In *Forty-second International Conference on Machine Learning*, 2025.

[50] G. Wang, K. Liu, J. Lin, G. Song, J. Li, and X. Han. Unimo: Unified motion generation and understanding with chain of thought. *arXiv preprint arXiv:2601.12126*, 2026.

[51] H. Wang, W. Weng, J. Wang, F. Zhao, G.-S. Xie, X. Geng, and L. Wang. Foundation model for skeleton-based human action understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[52] J. Wang, Y. Lai, A. Li, S. Zhang, J. Sun, N. Kang, C. Wu, Z. Li, and P. Luo. Fudoki: Discrete flow-based unified understanding and generation via kinetic-optimal velocities. *arXiv preprint arXiv:2505.20147*, 2025.

[53] Y. Wang, D. Huang, Y. Zhang, W. Ouyang, J. Jiao, X. Feng, Y. Zhou, P. Wan, S. Tang, and D. Xu. Motiongpt-2: A general-purpose motion-language model for motion generation and understanding. *arXiv preprint arXiv:2410.21747*, 2024.

[54] Z. Wang, X. Wang, S. Chen, Y. Cong, and M. Liu. Unimotion: A unified framework for motion-text-vision understanding and generation. *arXiv preprint arXiv:2603.22282*, 2026.

[55] W. Weng, X. Tan, X. Shu, G.-S. Xie, P. Zhou, and H. Wang. Bilingual text-to-motion generation: A new benchmark and baselines. *arXiv preprint arXiv:2603.25178*.

[56] W. Weng, H. Wang, J. Wang, L. He, and G.-S. Xie. Usdrl: Unified skeleton-based dense representation learning with multi-grained feature decorrelation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pages 8332–8340, 2025.

[57] W. Weng, X. Tan, J. Wang, G.-S. Xie, P. Zhou, and H. Wang. Realign: text-to-motion generation via step-aware reward-guided alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 10621–10629, 2026.

[58] B. Wu, J. Xie, M. Ding, Z. Kong, J. Ren, R. Bai, R. Qu, and L. Shen. Finemotion: A dataset and benchmark with both spatial and temporal annotation for fine-grained motion generation and editing. *arXiv preprint arXiv:2507.19850*, 2025.

[59] B. Wu, J. Xie, K. Shen, Z. Kong, J. Ren, R. Bai, R. Qu, and L. Shen. Mg-motionllm: A unified framework for motion comprehension and generation across multiple granularities. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27849–27858, 2025.

[60] T. Wu, Z. Fan, X. Liu, H.-T. Zheng, Y. Gong, yelong shen, J. Jiao, J. Li, zhongyu wei, J. Guo, N. Duan, and W. Chen. AR-diffusion: Auto-regressive diffusion model for text generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[61] Y. Wu, W. Ji, K. Zheng, Z. Wang, and D. Xu. Mote: Learning motion-text diffusion model for multiple generation tasks. *arXiv preprint arXiv:2411.19786*, 2024.

[62] Y. Wu, Y. Zhou, Z. Zhou, Y. Peng, X. Ye, X. Hu, W. Zhu, L. Qi, M.-H. Yang, et al. On the generalization of sft: A reinforcement learning perspective with reward rectification. In *International Conference on Learning Representations*, volume 2026, pages 27550–27571, 2026.

[63] Q. Yu, M. Tanaka, and K. Fujiwara. Remogpt: Part-level retrieval-augmented motion-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 9635–9643, 2025.

[64] W. Yuan, Y. He, W. Shen, Y. Dong, X. Gu, Z. Dong, L. Bo, and Q. Huang. Mogents: Motion generation based on spatial-temporal joint modeling. *Advances in Neural Information Processing Systems*, 37:130739–130763, 2024.

[65] J. Zhang, Y. Zhang, X. Cun, Y. Zhang, H. Zhao, H. Lu, X. Shen, and Y. Shan. Generating human motion from textual descriptions with discrete representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14730–14740, June 2023.

[66] J. Zhang, H. Fan, and Y. Yang. Energymogen: Compositional human motion generation with energy-based diffusion model in latent space. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

[67] M. Zhang, X. Guo, L. Pan, Z. Cai, F. Hong, H. Li, L. Yang, and Z. Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *IEEE/CVF International Conference on Computer Vision*, pages 364–373, 2023.

[68] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(6):4115–4128, 2024.

[69] M. Zhang, Y. Liu, S. Lin, Q. Dai, C. Luo, B. Guo, W. Jiang, P. Hou, A. Zeng, X. Yang, and X. Geng. Towards on-policy SFT: Distribution discriminant theory and its applications in LLM training. In *Forty-third International Conference on Machine Learning*, 2026.

[70] N. Zhang, Z. Li, K. W. Loh, M. Xu, Q. Wang, Z. Wen, X. He, W. Zhao, K. Gong, and M. Zhang. Dimo: Discrete diffusion modeling for motion generation and understanding. *arXiv preprint arXiv:2602.04188*, 2026.

[71] P. Zhang, P. Liu, P. Garrido, H. Kim, and B. Chaudhuri. KinMo: Kinematic-aware Human Motion Understanding and Generation. In *IEEE/CVF International Conference on Computer Vision*, 2025.

[72] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[73] Z. Zhang, A. Liu, I. Reid, R. Hartley, B. Zhuang, and H. Tang. Motion mamba: Efficient and long sequence motion generation. In *European Conference on Computer Vision*, pages 265–282. Springer, 2024.

[74] Z. Zheng, S. Jin, L. Liu, and M. Zhao. Next-scale autoregressive models for text-to-motion generation. 2026.

[75] B. Zhu, B. Jiang, S. Wang, S. Tang, T. Chen, L. Luo, Y. Zheng, and X. Chen. Motiongpt3: Human motion as a second modality. *arXiv preprint arXiv:2506.24086*, 2025.

[76] X. Zhu, B. Zhu, Y. Tan, S. Wang, Y. Hao, and H. Zhang. Enhancing zero-shot vision models by label-free prompt distribution learning and bias correcting. *Advances in Neural Information Processing Systems*, 37:2001–2025, 2024.

[77] X. Zhu, K. Zhao, L. Yi, S. Wang, Z. Wang, B. Zhu, H. Zhang, and X. He. Look carefully: Adaptive visual reinforcements in multimodal large language models for hallucination mitigation. In *International Conference on Learning Representations*, volume 2026, pages 81235–81256, 2026.

[78] X. Zhu, B. Zhu, Y. Li, J. Fang, S. Wang, K. Zhao, and H. Zhang. Hierarchical semantic alignment for image clustering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(34):29177–29185, 2026.

[79] X. Zhu, B. Zhu, S. Wang, J. Fang, K. Zhao, H. Zhang, and X. He. Principled steering via null-space projection for jailbreak defense in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22889–22899, 2026.

[80] Q. Zou, S. Yuan, S. Du, Y. Wang, C. Liu, Y. Xu, J. Chen, and X. Ji. Parco: Part-coordinating text-to-motion synthesis. In *European Conference on Computer Vision*, pages 126–143, 2025.

BiMoGen: Bidirectional Motion-Text Generation via Unified Masked Discrete Diffusion

Supplementary Material

This supplementary document contains additional visualizations, experimental details, ablation results, and sampling pseudo-code for BiMoGen. It is structured as follows. Sec. A presents additional qualitative results on text-to-motion generation and motion in-between. Sec. B provides further experimental details (Sec. B.1) and ablation studies (Sec. B.2), including the effects of semi-autoregressive sampling, backbones and classifier-free guidance, followed by the user study protocol (Sec. B.3). Sec. C presents the full pseudo-code of sampling with self-correction. We discuss the broader impact of the proposed method in Sec. D.

A More Visualization

In this section, we provide additional qualitative results to further demonstrate the effectiveness of BiMoGen. We compare BiMoGen with representative baselines on text-to-motion generation and motion in-between.

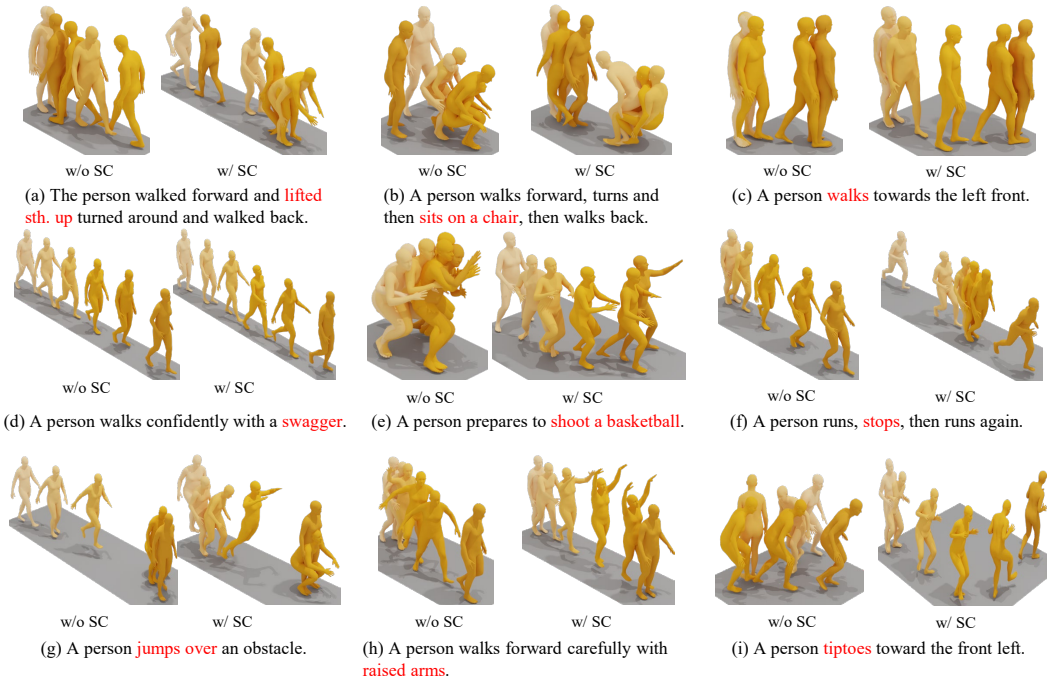

Figure S1: Qualitative comparison on text-to-motion generation. Red text marks descriptions that are not correctly reflected in the generated motion. BiMoGen produces motions that better align with the input prompts.

Visualization of Text-to-Motion Generation. Fig. S1 shows qualitative comparisons of text-to-motion generation on the HumanML3D test set. Compared with BiMoGen w/o SC, BiMoGen w/ SC produces motions that are more faithful to the input descriptions, especially in cases involving fine-grained semantics and directional cues.

Visualization of Motion In-between. Fig. S2 shows motion in-between results, where the start and end segments are fixed and the middle portion is generated. The masked-diffusion formulation supports this setting without architectural change, and BiMoGen produces smooth and semantically consistent transitions.

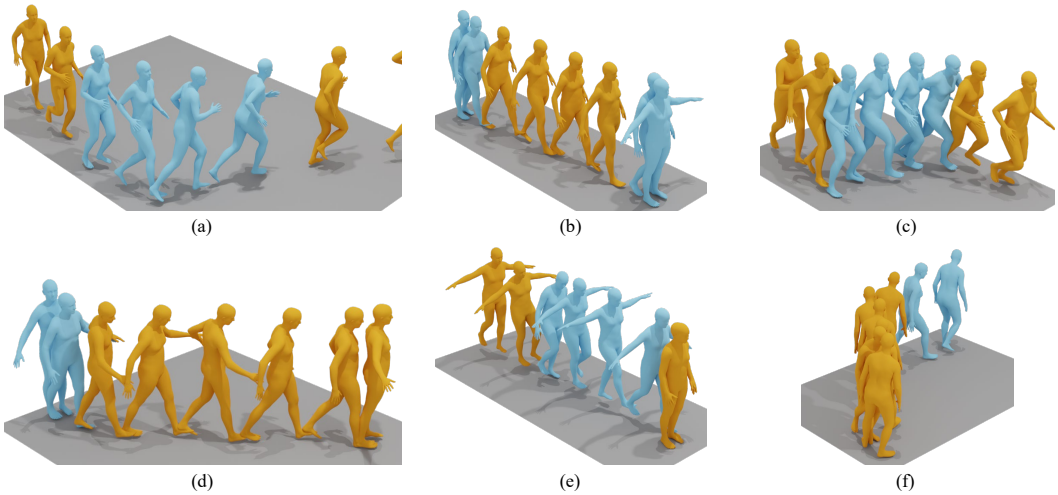

Figure S2: Visualization of motion in-betweening. Given prefix, suffix, middle, or start–end motion, BiMoGen performs motion in-betweening with smooth transitions. The human in blue represents the input, while the one in orange is generated by BiMoGen.

B Experiments

B.1 More Experiment Details

Metrics Definition. We follow the evaluation protocol used in prior text and motion generation work [13, 15, 14, 21, 75]. For T2M, we evaluate text and motion alignment, motion realism, and motion diversity. Unless otherwise noted, feature based metrics are computed with the official HumanML3D evaluator [13], using motion encoder $\phi(\cdot)$ and text encoder $\psi(\cdot)$. For M2T, we follow prior captioning evaluation [14] and adopt standard NLP metrics including BLEU [36], ROUGE-L [28], CIDEr [48], and BERTScore [72] to evaluate the fluency, relevance, and diversity of generated captions. We also report retrieval based R Precision for M2T to measure the alignment between generated texts and the corresponding motions.

Text and motion alignment. R@k measures retrieval accuracy in the shared evaluator space. For each query from one modality, candidates from the other modality are ranked by their evaluator distance, and the score is the fraction of samples whose paired item appears in the top k results. MM Dist measures the average distance between paired text and motion embeddings.

Motion realism. Fréchet Inception Distance measures the distributional distance between generated motions and ground-truth motions in the feature space.

Motion diversity. Diversity measures the variation among generated motions in the evaluator feature space.

Motion captioning. For M2T, generated descriptions are evaluated against reference descriptions. BLEU and ROUGE-L measure lexical overlap, CIDEr measures consensus with reference captions, and BERTScore measures embedding based semantic similarity. These metrics are used together because a single overlap metric does not fully capture caption quality.

B.2 Additional Ablation Study

We provide additional ablation studies on the design choices of BiMoGen.

B.2.1 Training Mixture

Table S1 compares T2M:M2T:M2M training ratios of 10:0:0, 8:1:1, and 8:2:0. T2M-only training (10:0:0) yields stronger T2M results than unified training (8:1:1), indicating a trade-off in supporting both generation directions. Removing M2M with an 8:2:0 mixture decreases R@1 from 0.555 to 0.522 and CIDEr from 60.2 to 58.8, while increasing FID from 0.069 to 0.133. Because the M2T

Table S1: Effect of training mixtures on HumanML3D. Ratios denote T2M:M2T:M2M; 10:0:0 is T2M-only.

T2M:M2T:M2M	R@1 ↑	R@3 ↑	FID ↓	MM Dist ↓	CIDEr ↑	BERTScore ↑
10:0:0	0.562	0.852	0.047	2.649	–	–
8:1:1	0.555	0.841	0.069	2.733	60.2	38.5
8:2:0	0.522	0.807	0.133	3.004	58.8	37.2

Table S2: T2M results with different bidirectional Transformer backbones on HumanML3D. Bert-Base and Bert-Large use pretrained weights, while BiMoGen is trained from scratch with the same training strategy.

Backbone	T2M						
	#Param	R@1↑	R@2↑	R@3↑	FID↓	Div→	MM Dist↓
Real	–	0.511	0.703	0.797	0.002	9.503	2.974
Bert-Base [8]	133M	0.544	0.737	0.829	0.063	10.177	2.792
Bert-Large [8]	366M	<u>0.553</u>	**0.747**	<u>0.839</u>	**0.057**	<u>10.018</u>	<u>2.734</u>
BiMoGen (Ours)	334M	**0.555**	<u>0.744</u>	**0.841**	<u>0.069</u>	**9.524**	<u>2.733</u>

proportion also changes, this comparison does not isolate M2M independently of the training mixture.

B.2.2 Backbone and Sampling Strategy

Robustness across backbones. Table S2 compares different bidirectional Transformer backbones under the same training recipe. The BERT variants are initialized from pretrained checkpoints, whereas BiMoGen is trained from scratch. Despite this difference, BiMoGen remains on par with BERT-Large and outperforms BERT-Base on most T2M metrics, with the best R@1, R@3, Div, and MM Dist. This result suggests that the proposed training strategy does not rely on language pretrained weights and can make effective use of model capacity across different backbone designs.

Effect of classifier-free guidance. Table S3 studies the effect of the CFG scale w for T2M generation. Increasing w from 1.0 to 4.0 consistently strengthens motion-text alignment and motion fidelity: R@1/R@3 improve from 0.505/0.795 to 0.551/0.838, FID decreases from 0.323 to 0.070, and MM Dist drops from 3.305 to 2.745. This confirms that guidance is important for masked bidirectional sampling, where the model must progressively select confident tokens under the textual condition. However, further increasing the scale to 5.0 no longer brings additional gains, slightly degrading R@1, R@3, FID, and MM Dist. We therefore set $w = 4.0$ as the default CFG scale, balancing text-motion alignment and motion quality.

Effect of semi-autoregressive sampling. Following Block Diffusion [1], Table S4 compares the standard parallel sampling of BiMoGen with semi-autoregressive sampling. In semi-autoregressive sampling, each block is denoised in parallel, while blocks are generated sequentially. The results show that parallel sampling consistently achieves the best performance across all metrics, suggesting that preserving global bidirectional refinement is more effective than imposing sequential block dependencies. Among the semi-autoregressive variants, smaller blocks perform better, whereas larger blocks progressively degrade both alignment and motion quality.

B.3 User Study Protocol

We further conduct a user study to assess the perceptual quality of generated motions. We sample 30 text descriptions from the HumanML3D test set and generate motions under the same prompts for all compared methods. We recruit 15 users and present the generated motions in random order. For the comparison with representative text-to-motion baselines, users are asked to select the best result among MotionLCM, MotionGPT, and BiMoGen in terms of motion-text alignment, physical fidelity, and coherence/fluency. For the SC ablation, users compare BiMoGen with and without SC, with an additional "Same" option when no clear preference is observed.

Table S3: Ablation on the CFG scale w for T2M generation. The number of sampling steps T is set to 20, and the self-correction is disabled.

CFG Scale w	Text-to-Motion					
	R@1↑	R@2↑	R@3↑	FID↓	Div→	MM Dist↓
1.0	0.505	0.693	0.795	0.323	9.977	3.305
2.0	0.535	0.729	0.827	0.144	10.001	2.840
3.0	0.550	0.**740**	0.829	0.088	9.884	2.753
4.0	**0.551**	0.739	**0.838**	**0.070**	9.870	**2.745**
5.0	0.543	0.739	0.833	0.076	**9.566**	2.760

Table S4: Effect of semi-autoregressive sampling for T2M generation. The first row denotes the standard parallel masked decoding used by BiMoGen. The remaining rows denote block-wise left-to-right semi-autoregressive variants, where tokens within each block are generated in parallel and different blocks are generated sequentially.

Decoding Strategy	Block Size	R@1 ↑	R@2 ↑	R@3 ↑	FID ↓	Div →	MM Dist ↓
Parallel sampling	–	**0.555**	**0.744**	**0.841**	**0.069**	**9.524**	**2.733**
Semi-Autoregressive sampling	1	0.534	0.721	0.827	0.181	9.415	2.915
	2	0.521	0.716	0.821	0.220	9.640	2.939
	4	0.521	0.717	0.818	0.336	9.534	2.985
	5	0.511	0.693	0.787	0.527	9.435	3.040
	10	0.496	0.681	0.776	1.018	9.332	3.148

C Sampling with Self-Correction

Algorithm 1 gives the pseudo-code of sampling with self-correction. Given condition tokens x^{cond}, BiMoGen iteratively refines a fully masked target: at each step, the model predicts clean target tokens under classifier-free guidance, and $\mathrm{Remask}(\cdot)$ then remasks the lowest-confidence predictions following LLaDA [35]. Remasking is restricted to positions masked at the start of the current step, so tokens unmasked earlier are preserved. Here, x, s, t, and T denote the current state, prediction confidences, current step, and total number of steps. At steps in R, self-correction completes the current target, then revises the visible tokens, leaving the remaining masked tokens to later steps.

Algorithm 1 Iterative Masked Sampling with Self-Correction

Require: Condition x^{cond}, target length L, model p_θ, steps T, CFG scale w, correction steps $\mathcal{R}$
Ensure: Generated target tokens $\widehat{x}$

$x \leftarrow [M]^L$
for $t = 1$ to T **do**
  $P^{\mathrm{cond}} \leftarrow p_\theta(\cdot \mid x^{\mathrm{cond}}, x), \quad P^{\mathrm{ucond}} \leftarrow p_\theta(\cdot \mid \emptyset, x)$
  $P \leftarrow wP^{\mathrm{cond}} + (1-w)P^{\mathrm{ucond}}$
  $y_i \leftarrow \arg\max_v P_i(v), \quad s_i \leftarrow \max_v P_i(v)$ for i with $x_i = [M]$
  if $t \in \mathcal{R}$ **then**
    $\widetilde{x} \leftarrow x$
    $\widetilde{x}_i \leftarrow y_i$ for i with $x_i = [M]$ ▷ temporary complete target
    $Q \leftarrow p_\theta(\cdot \mid x^{\mathrm{cond}}, \widetilde{x})$
    $x_i \leftarrow \arg\max_v Q_i(v)$ for i with $x_i \neq [M]$ ▷ revise committed tokens
  $\bar{x}_i \leftarrow y_i$ for i with $x_i = [M]$, and $\bar{x}_i \leftarrow x_i$ otherwise
  $x \leftarrow \mathrm{Remask}(\bar{x}; x, s, t, T)$ ▷ remask low-confidence predictions
$\widehat{x} \leftarrow x$
return $\widehat{x}$

D Broader Impact

Broader impact. This work may benefit motion content creation, animation, virtual agents, and motion retrieval through bidirectional language-motion generation. Since the model produces abstract motions rather than identifiable visual appearances, its direct risk is limited. Still, generated motions may be used in downstream synthetic avatar or animation systems, so responsible use, consent for real motion data, and disclosure of synthetic content should be considered in deployment.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main contributions, including the unified masked diffusion framework, decoupled training, and self-correction. The claims are supported by the experiments in Sec. 4 on HumanML3D and KIT-ML.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the main limitations in Sec. 5, including the dependence on VQ-VAE tokenization and the lack of explicit fine-grained body-part control.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present formal theoretical results, theorems, or proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sec. 4 reports the datasets, evaluation protocol, model configuration, training schedule, optimizer, sampling settings, and compute device used for the main experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We do not currently provide full reproduction scripts.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Sec. 4 specifies the datasets, metrics, model architecture, optimizer, learning rates, training epochs, task mixture ratio, condition dropout, and inference settings.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: For the T2M and M2T experiments, we follow the standard evaluation protocol and report averaged results over 20 evaluation runs, rather than single-run point estimates.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Table 4 reports parameter counts, FLOPs, and latency for the main model variants, and Sec. 4 states that experiments are run on a single NVIDIA H100.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses standard public motion-language benchmarks and does not involve new human-subject data collection. We have reviewed the NeurIPS Code of Ethics and believe the research conforms to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses positive impacts on motion generation, animation, virtual agents, and assistive content creation, and also notes risks related to synthetic content misuse, dataset bias, and safety critical deployment.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk scraped datasets, pretrained language models, image generators, or other assets requiring special release safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper cites the existing datasets, models, and evaluation assets used, but the current draft does not explicitly list their licenses, versions, or terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce or release new datasets, benchmarks, or model assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [No]

    Justification: Appendix B.3 summarizes the user study protocol. Participants received no monetary compensation. Full participant instructions are not included.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work does not involve new crowdsourcing or human-subject studies, so IRB or equivalent approval is not applicable.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as an important, original, or non-standard component of the core method, nor are they used to generate data, labels, or evaluation results.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.